JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 3, 2022, p. 3143-3151
https://publishoa.com
ISSN: 1309-3452

Development of Graphical User Interface (GUI) for Teaching Braille in Class for The Blind People

[1]Surya Dharshan s/o Surash, [1]Shaharil Mad Saad, [1] Wan Aliff Abdul Saad, [1] Muhammad Danial Abu Hasan, [1] Zair Asrar Ahmad, [2] Mohd Azwarie Mat Dzahir

[1]*School of Mechanical Engineering, Faculty of Engineering, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia*

[2] *Faculty of Built Environment & Surveying, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia*

shaharil@utm.my

ABSTRACT – Assistive technology is crucial to ease the teaching, learning, reading, and writing process for visually impaired students. In a cluster braille teaching class, visually impaired students often require a tool to have a bi-directional communication with their teacher as it is hard and not systematic for them to convey their information to teacher in a class through their voice. It is also certainly not easy to teach a blind student in Braille especially when teaching is done for a group of blind students. This is because teachers for blind students must be expert in Braille which limits the number of educators available for blind student. Therefore, the objective of this thesis is to design and develop a Graphical User Interface (GUI) that can convert and display grade 1 braille inputs into alphabets, numbers, and punctuation and vice versa which will allow teachers who does not know Braille to communicate and teach the blind students. The developed braille translating GUI must be able to send and receive data from both teacher and student's Raspberry Pi wirelessly which will allow two ways communication between blind students and their teacher. GUI for student must have feature to type message in Braille. GUI is to be developed in Python programming language. An experiment is conducted to perform test runs on three algorithms that the GUI is based on which are Braille translation, Braille typing and bi-directional communication. All results show positive outcomes. Grade 1 Braille translation (back and forth) was correct but capital letters are ignored. Algorithm created to type braille in student's GUI worked accordingly, however, keyboard with N-key rollover feature is strongly recommended for braille six key input. Finally, bi-directional communication based on socket and multithreading programming enables proper communication between teacher's and student's Raspberry Pi via Wi-Fi.

Keywords - Braille, Raspberry Pi, Python, Bi-directional communication, Braille translation, Braille six key input.

I. INTRODUCTION

In this modern, digital, and informational age, we have seen technology has advance significantly in every sector. There has been a steady growth of advancement in developing assistive technologies for braille. Example of assistive technology created for braille is such as Perkins Brailler (standard and smart version), braille embosser, refreshable braille display, screen reader, slate and stylus and portable braille writer (Jot a Dot, Tatrapoint and Mountbatten brailler). However, technological aid in Braille for blind people is still complicated and considered underdeveloped or often gets overlooked while it is mostly unreachable for many of them in this world especially in teaching and learning factor. The International Council for Education of People with Visual Impairment estimates that only 10% of children who are blind in developing countries have access to education or tools for writing braille [1].

The problem of lacking in technological aid in Braille teaching and learning sector is a serious matter. A visually impaired student cannot use sight to gain and learn information. Thus, other sense like touch and hearing is the only way for them to learn. This poses the main difficulty where educators need to use special ways to teach them.

In a teaching and learning process, interactions between student and teachers are vital to keep the class active and interactive. This way allows teachers to make sure students can keep up and understand whatever contents are being taught. Thus, communication between each other is important. Since visually impaired students cannot utilise the sense of sight, they rely heavily on sense of touch and hearing to send and receive information. Teachers could always teach and transmit information using their voice to them. While it is a plausible way, it often becomes a one-way communication especially when the teachers are teaching a group of students. As much as they hear, visually impaired students also would like to voice out their questions or opinion in the class, but it would be hard for a single teacher to hear all of them. Thus, a medium or some sort of "assistive technology" is needed here to be the bridge where blind students can use it to communicate or transmit information with their teachers. A system which allows teachers to send and receive information systematically with visually impaired students regardless of how many of them as well as the students to be able to convey information in Braille with their teacher is what is required here. This is the highlighted issue and the main concern of this project.

A system that can translate braille into alphabets, numbers and punctuations vice versa can be very handy in teaching and learning process as any teacher who either fluent or not fluent in braille can use it to teach blind students. In addition, a language converting system that can also transmit information both ways in real time with other same kind of its system within a seamless network should not a be a pipe dream as it is indeed possible to make it in this technologically advanced world. It is also possible to develop this system for an affordable price range and for convenient usability. Every problem has its unique solution, and this project aims to tackle this problem to create a better future and learning environment for every visually impaired student.

II. LITERATURE REVIEW

Braille is a system of touch reading and writing for people who are visually impaired [2]. It consists of raised dots which represents either letters of alphabets, numbers, or punctuation as well as symbols to show letter grouping. Braille is read by moving the hand and index fingers from left to right along each line passing by touching each raised dot in the line. This way of Braille system utilises their touch sensory by feeling each raised dot to understand and transmit information or knowledge in it. Braille is traditionally written in embossed paper. Braille system was invented by Louis Braille in the early 1800s. The inspiration for Louis Braille to create this efficient written communication system (Braille) for blind individuals came from Charles Barbier's 'night writing' [2]. Louis Braille improved the raised dots system now known as Braille so that a fingertip could cover the entire cell unit with one impression and move rapidly from one cell to the next. Till this day, it remains as how he invented it.

Braille system has been adapted to many languages. Braille code generally consists of patterns of raised dots arranged in cells up to 6 dots in 3 × 2 configuration (3 row, 2 column) [3]. There are 64 possible combinations for each Braille cell which is obtained by altering the presence of the 6 dots [4]. Each Braille cell represents a braille letter, numeral, symbol, or punctuation mark. There are 3 version of Braille code [3]. Grade 1 is for beginners who just started reading braille. Grade 2 is same as Grade 1 with contraction of words added (the most used). Grade 3 is for unofficial writings as this version makes entire words shortened to a few letters only. After decades of inventions, braille is still widely used till this day mostly by visually impaired people. Braille system has given literacy to many people for decades. Blind people can now perceive all knowledge and information in printed words there is to offer just like everyone else. The effect is profoundly rewarding and helps them to achieve school and career success.

However, teaching in braille for blind students has never been an easy task. Technological aid in braille teaching and learning is vital because with the absence of sense of sight, blind student must use their other senses like touch and hear to gain information and knowledge. This poses a whole new series of problems that requires innovative technologies to mitigate those issues. While tools like braille typewriter, braille display and screen reader do exist to be helpful, many researchers are actively exploring on ways to make the fundamental processes of teaching and learning in Braille to be more approachable and effective. In this study, the highlights of the matter will be on Braille teaching methods, Braille translating methods and GUI Implementation in teaching. The table 1 below provide background information about some of the past research that has been explored in braille teaching methods, braille translating method and GUI Implementation in teaching.

Table 1: Teacher GUI's Functions

Authors	**Assistive technology for teaching in Braille**
[4]	eBraille software in GUI with Braille Display to read Al-Quran
[5]	'Braille Box' for blind student to understand GUI in computers
[6]	Smart glove as sensor and tactile feedback with microcontroller translator
[7]	Image processing using MATLAB to transcribe Braille code representing Odia language into Odia word as text
[8]	Bluetooth and Arduino controlled solenoid actuators for tactile interface to teach Braille code
[9]	Electronic Braille training system to teach Grade-1 Braille English alphabet

Overall, although yielding great results, the past research studies still lack in some criteria. Thus, the issues that are extracted based on the three categories of the past research studies and will be focused on this project. The issues are as following:

- Assistive devices capable of teaching a group of students are limited to one way communication.

- Bidirectional communication enabled assistive devices were in a form that is not practical to be used in teaching in cluster.
- GUI usage still requires a form of medium for blind people to interact with.
- The used field of technologies exhibit only minor future proofing in terms of technology upgrading pathways.

To work around all those issues and to introduce modern technologies in the field of Braille, Raspberry Pi and Python programming language has been selected for hardware and software respectively for this project. Raspberry Pi is a single board computer. It is like Arduino microcontroller but more powerful has more functionalities [10]. Just like a desktop computer, Raspberry Pi has onboard components such as a processor, RAM, HDMI, audio output and USB ports and it also runs a fully-fledged operating system (OS) based on Debian Linux, called Raspberry Pi OS. The reason it is suitable for this project is because of the following reasons:

- Affordable, portable and provide adequate computing power.
- Raspberry Pi OS works well with python programming language.
- Offers onboard Wi-Fi connectivity which is crucial for this project.
- GIPO headers available on this board allows different electronic circuitries to be plugged directly to the board. This feature is particularly useful if this project is intended to be further developed with a braille typewriter and refreshable braille display that might can utilize well these GIPO headers.

Next, Python programming language with complex semantics is an interpreted, object-oriented, high-level programming language [11]. Its high-level built-in data structures, along with dynamic typing and dynamic linking, make it highly desirable for Rapid Application Development, as well as for usage as a scripting or glue language to connect existing components together. Python also provide various choices to develop Graphical User Interfaces (GUIs). Tkinter, a standard Python GUI toolkit, will be used in this project to develop the required GUI because the functionalities for Tkinter are well documented.

GUI obviously cannot be directly usable for blind students because it requires eyesight to operate the program. However, there are other few reasons on how GUI can be beneficial for this project. Firstly, it allows teachers to have better connectivity with individual students, because, in terms of braille translation capabilities, GUI is far more convenient to be used compared to manual ways of translating braille through referring braille translation charts or books. Secondly, GUI allows developed coding algorithm to be more interactive and many features and functions can be packed in a single GUI. Thirdly, even though braille GUI cannot be used visually by braille students, GUI coding algorithm can be tweaked in a way it can be beneficial in some other ways for them. For example, braille-six-key-input concept for braille typing, words to speech recognition functionality and braille translation ability can be integrated to the GUI. Not only it can help blind students to navigate through the GUI, lecturers guiding the blind students can also monitor the students' progress and keep up to date with them (by sending messages) through the GUI too.

III. METHODOLOGY

A. *Proposed Method*

The general idea of this project is to develop a GUI that can translate Braille into text and vice versa that can be operated in a single board computer using latest technology available today. This setup must be able to be implemented for the usage of bidirectional communication between blind students and their teachers in a class. This project idea should also provide a low cost, less complex, high reliable and user friendly to implement solution for the mentioned use case.

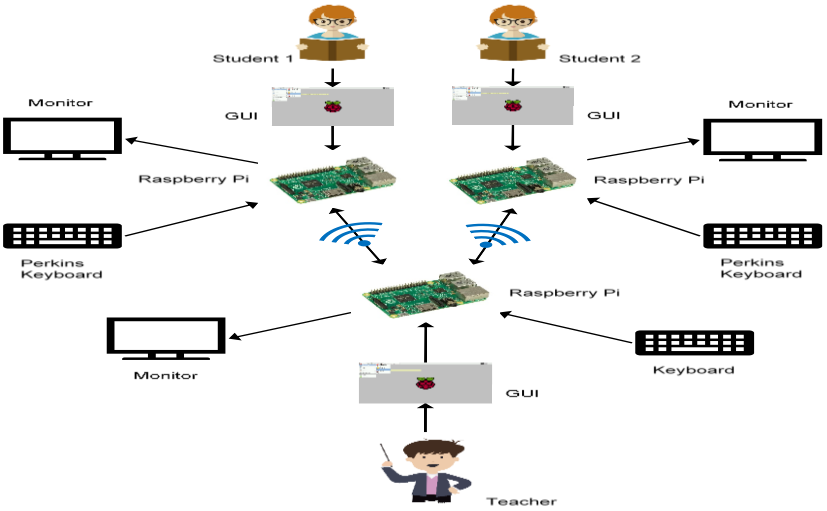


Fig.1: Overall System

Proposed approach is conceptually simple. GUI will be developed using Tkinter Python programming language. The developed GUI program will then be uploaded to Raspberry Pi single board computer. The Raspberry Pi will be connected to peripherals like mouse and keyboard for data input and monitor for visual output. If GUI program is written correctly, translation of Braille into text and vice versa can be done at this stage. Next, to simulate communication between a teacher and blind students, 3 Raspberry Pi will be implemented. Each Raspberry Pi will have the GUI program with the appropriate coding programs to send and receive information from each other. One Raspberry Pi will be set as the host (teacher) while the other 2 will be set as user (blind students). Student's Raspberry Pi will be connected under teacher's Raspberry Pi's Wi-Fi (offline network). All 3 Raspberry Pi will have the mentioned peripherals for information input and output. Setup is as shown in Fig.1. GUI on teacher's Raspberry Pi will be configured to be able to translate incoming data from students' Raspberry Pi which is in Braille format into alphanumeric format and be able to send information to any selected user Raspberry Pi. Students' Raspberry Pi (User 1 and User 2) will also have the mentioned peripherals for information input and output. Supposedly, students' input and output peripheral should be a braille typewriter and a refreshable braille display, respectively. However, to simplify this project, keyboard and monitor will be used to simulate the braille typewriter and a refreshable braille display. Students' keyboard will be configured to be able to type in braille (configured into Perkin keys, SDFJKL, also known as Braille six key input) and monitor will simulate the reading of braille through touch in visual format. The ability for student's QWERTY keyboard to be able to type in Braille will be configured in student's GUI development.

In this case, the use of monitor can help us track the GUI progress much better. In proper real-life application, with some extra coding algorithm, refreshable braille display can be included in the GUI system for blind student to perceive information through sense of touch while teacher can monitor their progress in each of their monitors. In terms of hardware, integration of refreshable braille display with Raspberry Pi can be done using GIPO headers available on Raspberry Pi. GUI on students' Raspberry Pi will be configured to be able to read, write, and send information back to teacher's Raspberry Pi in Braille format only (to ease student's GUI use case).

B. Function and Features of GUI

Function and features of GUI must be determined first before designing the interface itself. This step will be helpful because limits and capabilities of the GUI in either characters translation or in message transmission can be clarified. GUI is designed according to teacher and student use case. Therefore, there will be two distinctive GUI design (teacher and student). GUI will be designed with the following core features in mind:

- GUI is capable of Braille to alphanumeric and punctuation translation and vice versa.
- GUI only able to utilise Grade 1 Braille System based on Braille ASCII.
- GUI allows connectivity to other Raspberry Pi through its onboard Wi-Fi.
  - Teacher's Raspberry Pi will be configured to be used as wireless access router.
- Network between each Raspberry Pi will be centralized.
  - Teacher's GUI will be the server; thus, it will have server controlling functionalities as well as the ability to monitor connected clients (students).
  - Teacher's GUI can either send message to specific student or broadcast to all of them.
  - Student's GUI can only send message to teacher's GUI.
- Braille typing in student's GUI is configured based on braille six key input
- GUI for both parties is able display incoming and outgoing messages.

C. Designated Keys for Braille Input

Each cell has 6 dots. In this system, each dot is assigned a key in the keyboard. The assigned keys are shown in the following Figure. The keys SDFJKL are chosen because there is a small ridge at key F and J on the keyboard. This ridge acts as a reference point that allows user to place their index finger for both hands at F (left finger) and J (right finger) key without need to look back at keyboard [4]. The rest of the keys SD for left hand fingers and KL for right hand fingers follows respectively. If a key is pressed, the dot will be darken as shown in the figure. There are several other keys that will be implemented for its respective use case which are listed below:

Space bar:	to enter space between words
Enter:	to send the message
Backspace:	to delete each braille cell

Using the keyboard hook method proposed in Blenkhorn research study, an algorithm for Braille typing with Braille six key input can be developed in Python [12].

D. Communication between Raspberry Pi

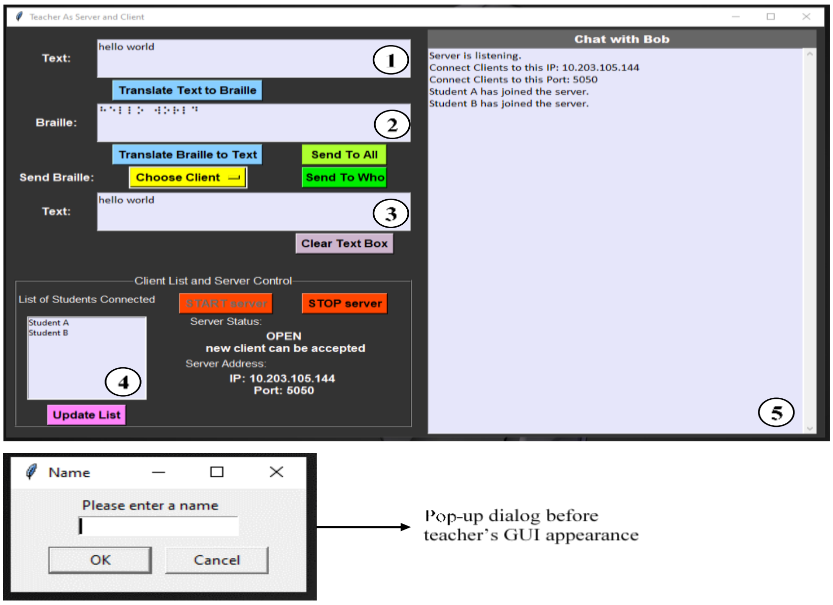
Three Raspberry Pi in this system design requires a method to set-up a network and be able to transmit information back and forth (information in this case is the messages being sent and received through the GUI between students and teacher). The network is a wireless communication through Wi-Fi, and it is offline. This means this set-up will not be requiring an internet connection for the communication to work. Student's Raspberry Pi will be connected to the onboard Wi-Fi of teacher's Raspberry Pi. In another words, teacher's Raspberry Pi will be the server while Student 1 and 2 's Raspberry Pi will be the client. Network will be configured so that only teacher and student can communicate to each other. There will be no line of communication between students. Thus, the network between each Raspberry Pi will be centralized. To enable these connections between the Raspberry Pi, python low-level networking interface will be used in this project. Specifically, socket and multithreading programming will be utilized.

Socket programming allows communication between two programs or between two devices by creating socket for each of them and establish connections either in server-client model or client-client model (client-server model will be used for this project). A socket is defined as one endpoint of a two-way communication link between two programs running on the network [13]. Endpoint is a combination of an Internet Protocol (IP) address and a port number. Every socket formed is assigned to an IP address and a unique port number so that TCP layer can identify the application to which data is being delivered. Socket programming requires multiple process to be executed concurrently. However, all the process cannot be run on the main program thread or else the GUI will be unusable. To alleviate this issue, multithreading programming is included. Multithreading can execute multiple threads simultaneously in a single process [14]. Thread is an independent unit of execution generated within the context of a running process or application. Integrating multithreading with the socket programming also allows server to handle multiple clients.

IV. RESULT & DISCUSSIONS

This section will explain the finalized, developed GUI for both teacher and student. Fig. 2 displays the final version of GUI developed for teacher and Fig. 3 is for student. Table 2 explains functionalities included in teacher's GUI while Table 3 explains functionalities included in student's GUI.

To validate the functionality of the developed algorithm and GUI to be working properly, an experiment was set up based on the overall system. Using the setup, several test runs were initiated. These test runs were focused on validating the working process of the GUI in bi-directional communication, Braille translation, and Braille typing.



hello world

hello world

Server Status:
OPEN
new client can be accepted
Server Address:
IP: 10.203.105.144
Port: 5050

Student A
Student B

Server is listening.
Connect Clients to this IP: 10.203.105.144
Connect Clients to this Port: 5050
Student A has joined the server.
Student B has joined the server.

Please enter a name

Pop-up dialog before
teacher's GUI appearance

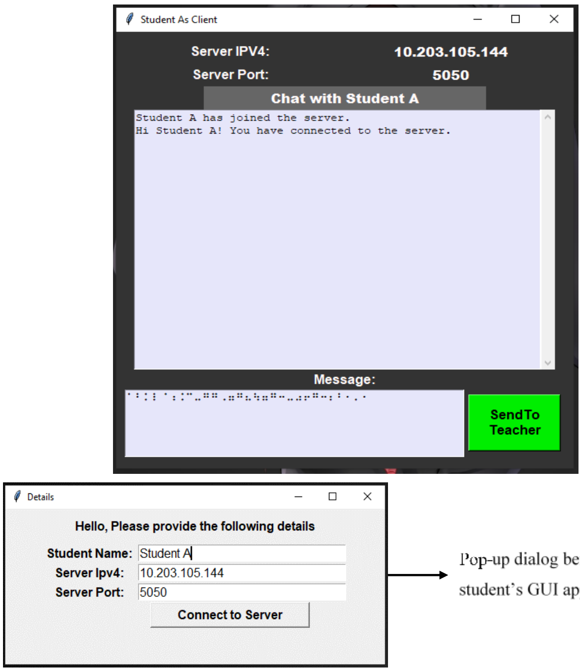
Fig.2: Teacher's GUI

Table 2: Teacher GUI's Functions

Widget	Function
Translate Text to Braille	Translates alphanumeric text to Braille format.
Translate Braille to Text	Translates Braille to alphanumeric text.
Choose Client ⊐	Drop menu to choose connected students to send message.
Send To Who	To send messages to student chosen in choose client drop menu.
Send To All	To send messages to all students connected to teacher's GUI server.
Clear Text Box	To clear translation text box.
Update List	To check on connected students and update choose client drop menu.
START server	To start server to accept new clients to connect to the teacher's GUI server.
STOP server	To stop server from accepting new clients to connect to the teacher's GUI server.
Text Box ①	To insert text in alphanumeric format.
Text Box ②	To display translated text in Braille format.
Text Box ③	To display translated Braille into text format.
Text Box ④	To display students in connection with teacher GUI server.
Text Box ⑤	To display incoming and outgoing messages as well as notifications.

A. *Bi-directional Communication Test Run*

Validation for bi-directional communication is performed by visually inspecting the message displaying textbox in both teacher and student's GUI. The bi-directional communication test run was conducted 10 times to obtain a consistent outcome. Operations tested in teacher's GUI are about sending messages to specific student or to broadcast to all of them in Braille format and receiving messages



```
Student A has joined the server.
Hi Student A! You have connected to the server.
```

[illegible]

Student Name: Student A
Server Ipv4: 10.203.105.144
Server Port: 5050

Pop-up dialog before student's GUI appearance

Fig.3: Student's GUI

Table 3: Teacher GUI’s Functions

Widget	Function
	To send messages to teacher. (press ‘Enter’ in keyboard to active it)
Big text box	To display incoming and outgoing messages
Small text box	To write in braille using F, D, S, J, K, and L keys Can use backspace to delete braille characters and spacebar to add gaps between words.

from student’s GUI who are connected to teacher GUI’s server. Operations tested in student’s GUI are about sending messages to teacher in text format and receiving messages from teacher in Braille format. Based on those operations, the results show that the developed algorithm in the GUI for this purpose was successful. In addition, text to speech conversion was added to student’s GUI. While students reading, the conversion can help students to listen to incoming and outgoing messages more effectively. Besides, the teacher’s GUI server was also tested for more than two client connections. It was found that teacher’s GUI’s server indeed able to handle multiple clients (student GUI). This result is substantial because it confirms that this concept of GUI does certainly applicable for braille cluster teaching. However, it was noted that with each client connection added, performance penalty will be more significant as all the threads generated will be sharing a finite amount of hardware resource available in Raspberry Pi. Other than that, this test run also demonstrates the benefits of using the developed algorithm that make use of teacher’s Raspberry Pi’s onboard Wi-Fi to act as hotspot allowing students to establish connection and communicate with the teacher without the need for Internet access.

B. Braille translation Test Run

The Braille translation test run was conducted 10 times to obtain a consistent outcome. Validation for Braille translation is performed by visually inspecting the translation of each character from Braille to text and vice and versa.

In this test run, all characters inserted were translated correctly. By utilizing dictionary function with for-loop and if statements in Python, each character is linked to its corresponding character in other format. In this case, each alphanumeric and punctuations is linked to its corresponding Braille code. This method is feasible because dictionary function holds key: value pair. When a known key is called, it retrieves the value stored for the specific key. However, the developed Braille translation algorithm do have some limitations which were, capitalization of letters is ignored, and it is only based on braille grade 1.

C. Braille Typing Test Run

The Braille typing test run was conducted 10 times to obtain a consistent outcome. Braille typing is available only in student’s GUI. Validation for this part is done by inspecting which braille codes that can be typed and which one cannot. Braille codes that can be typed will be displayed in the small textbox in student’s GUI. If the right braille is displayed, it means the combination of keys pressed and released is detected and converted successfully.

As it can be observed in above table, most alphanumeric and punctuations with 1, 2 or 3 keys combination to form braille works successfully. However, only some 3, 4 or 5 keys combination to form braille works successfully. 6 keys combination to form braille of equal symbol did not work. Out of 63, 8 Braille inputs was not successful. There are several deductions that can be proposed as for why the results is obtained in that manner. The most likely cause for the failed key detections is because of the lack of N-Key Rollover feature. N-Key Rollover feature is the ability of the keyboard to correctly recognize multiple key presses at once [15]. The test run was performed using cheap rubber dome style keyboard which does not have N-Key Rollover feature. The test was performed again with an external mechanical switch keyboard which has N-key rollover feature and the output results showed significant improvement for braille that require more than 3 keys combinations. The findings about Braille typing mechanism and the results of this test runs obtained in this project is similar with the paper written by Paul Blenkhorn [12]. This concludes that the deductions made about the significant of N key rollover feature in braille six key input in this analysis is proven correct. In summary, from this test run, it can be considered that the developed braille typing using the braille six-key-input method was successful as most of the Braille inputs were correctly detected and displayed. However, the algorithm for Braille typing can be refined further to mitigate the issue raised by errors like Braille inputs not being detected as mentioned previously. Plus, this issue can also be fixed by using keyboards that has N-Key Rollover feature.

V. CONCLUSIONS

This study was carried out with the purpose to design and develop a GUI for teacher and blind student which can do braille translation, bi-directional communication and braille typing (braille typing in student’s GUI only).

Based on the conducted test runs for the developed GUI, all three algorithms worked excellently. Both the teacher's and the student's GUI can accurately convert braille characters into alphabets, numerals, and punctuation, and vice versa. However, this translation algorithm has two limitations: it cannot convert capital letters and it is only based on braille grade one level. Conducted test runs for braille typing shows each braille character can be written well using ‘S’, ’D’, ’F’, ’J’, ’K’, and ’L’ keys in standard QWERTY keyboard. The results also shows that keyboards must have N-key rollover feature to enable multiple key presses at once which is essential for braille six key input.

With the aid of Raspberry Pi, test runs for bi-directional communications performed in both teacher’s and student’s GUIs, which were operated on their own Raspberry Pi, yielded encouraging results. Teacher’s GUI was able to deliver Braille messages to individual students or broadcast to all of them. It was also able to receive messages from students which were already pre-translated automatically from Braille to alphanumeric before being sent. On student’s GUI side, braille messages from teacher are correctly received while braille message written by student is also properly being sent in alphanumeric format to the teacher. Every outgoing and incoming message in student’s GUI, as well as each braille character that is typed is converted into speech as audio output for students which can be helpful for them. All transmissions were completed successfully via wireless communication (Wi-Fi) on Raspberry Pi, which did not even require Internet connection.

This project has demonstrated how beneficial Python programming language and the Raspberry Pi can be in building a GUI that is to be utilized as a strong teaching medium in a class for both teachers and blind students. There are several ways this project can further be improved and improvised. The suggestions are as following:

A. *Integration of braille display via GIPO headers in Raspberry Pi*

This project setup is only a simplified version. For proper practical application, braille display should be included for blind students to read incoming and outgoing messages by touch. Since Raspberry Pi is used in this project, this setup can further be extended with ease. The GIPO headers in Raspberry Pi allows easy integration of any kind of electronics especially braille display in this case.

B. *Upgrading Braille translation*

Currently, the developed braille translation imposes two known major limitations which were, capital letters are ignored, and translations is only based on braille grade 1 level. Tackling these issues will make the braille translation more versatile and this can be done using various functions and modules available in Python.

C. *Port forwarding*

This project setup emphasizes on offline operation. This is advantageous because the developed GUI software does not require an internet connection as it utilizes local wireless access router (teacher’s raspberry pi) to connect to other devices (student’s Raspberry Pi). However, if internet connectivity is incorporated, connections between student’s and teacher’s Raspberry Pi can be established regardless of the distance between both devices. This can be accomplished using the method of port forwarding. Port forwarding allows a computer in a network to be accessible by other computers on the internet, even though they are behind a router [16]. By this way, student’s Raspberry Pi does not require to be on the same network or under teacher’s Raspberry Pi’s network. Students can simply establish connection with their teacher over the internet from wherever they are, and it will be very convenient for everyone to accomplish various task such as file sharing, online meetings, assignment submissions and instant messaging. This approach can be very useful especially during pandemic seasons, when teachers and students are unable to gather and meet up in a classroom as usual.

D. *Improving Text-to-Speech Converter*

The existing text to speech functionality included in student’s GUI has a few drawbacks. Most notably, the text to speech conversion makes the whole student’s GUI operation seems lagging. Plus, the support for text to speech conversion in Raspberry Pi OS is not properly refined compared to Windows OS. Improving this functionality can certainly boost the usage effectiveness for the blind student.

ACKNOWLEDGEMENT

The authors would like to acknowledge the financial sponsorship provided by Universiti Teknologi Malaysia (UTM) through university research grant, No: 4C489, 16J14 and 20J83.

REFERENCES

[1] Tim Connell, “The Challenge of Assistive Technology and Braille Literacy,” *American Foundation for the Blind*, 2008. https://www.afb.org/aw/9/1/14277.

[2] “History of Braille,” *Braille Works*. https://brailleworks.com/braille-resources/history-of-braille/#:~:text=Braille is a system of,to right along each line.

[3] “Braille Alphabet,” *Braille Works*. https://brailleworks.com/braille-resources/braille-alphabet/.

[4] F. Razaly *et al.*, “Dedicated Graphical User Interface system for the visually impaired users in learning Al-Quran,” *Proc. 9th WSEAS Int. Conf. Appl. Comput. Eng. ACE ’10*, no. January, pp. 210–213, 2010.

[5] N. Sribunruangrit, C. Marque, C. Lenay, and O. Gapenne, “Graphic-user-interface system for people with severely impaired vision in mathematics class,” *Annu. Int. Conf. IEEE Eng. Med. Biol. - Proc.*, vol. 26 VII, pp. 5145–5148, 2004, doi: 10.1109/iembs.2004.1404432.

[6] T. Choudhary, S. Kulkarni, and P. Reddy, “A Braille-based mobile communication and translation glove for deaf-blind people,” *2015 Int. Conf. Pervasive Comput. Adv. Commun. Technol. Appl. Soc. ICPC 2015*, pp. 1–4, 2015, doi: 10.1109/PERVASIVE.2015.7087033.

[7] K. Parvathi, B. M. Samal, and J. K. Das, “Odia Braille: Text transcription via image processing,” *2015 1st Int. Conf. Futur. Trends Comput. Anal. Knowl. Manag. ABLAZE 2015*, no. Ablaze, pp. 138–143, 2015, doi: 10.1109/ABLAZE.2015.7154983.

[8] K. Kavitha, A. Privadarshini, and V. Saradha, “Braille Teaching System for the Visually Impaired,” *Proc. 2nd Int. Conf. Electron. Commun. Aerosp. Technol. ICECA 2018*, no. Iceca, pp. 1301–1304, 2018, doi: 10.1109/ICECA.2018.8474928.

[9] P. J. Osuch and S. Sinha, “An electronic solution to automate the process of Grade-1 braille training,” *Proc. 3rd IEEE Glob. Humanit. Technol. Conf. GHTC 2013*, pp. 47–52, 2013, doi: 10.1109/GHTC.2013.6713652.

[10] Richard Saville, “What Is a Raspberry Pi?,” *Lifewire*, 2020. https://www.lifewire.com/what-is-raspberry-pi-4042337#:~:text=Why Is It So Popular,t the only factor here.

[11] “What is Python? Executive Summary,” *Python Software Foundation*. https://www.python.org/doc/essays/blurb/.

[12] P. Blenkhorn and G. Evans, “Six-In - braille input from a QWERTY keyboard,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3118, pp. 484–489, 2004, doi: 10.1007/978-3-540-27817-7_71.

[13] “What Is a Socket?,” *ORACLE*. https://docs.oracle.com/javase/tutorial/networking/sockets/definition.html (accessed Aug. 12, 2021).

[14] B. Burns, “What Is Multithreading: A Guide to Multithreaded Applications,” *Perforce Software*, 2020. https://totalview.io/blog/multithreading-multithreaded-applications (accessed Aug. 12, 2021).

[15] “ControlSpace: N-Key Rollover: What It Is and How To Test Your Keyboard.” http://blog.controlspace.org/2010/08/n-key-rollover-what-it-is-and-how-to.html (accessed Aug. 12, 2021).

[16] J. Bauer, “The Ultimate Guide to Forwarding Ports in Your Router,” *Port Forward*, 2021. https://portforward.com/how-to-port-forward/ (accessed Aug. 12, 2021).